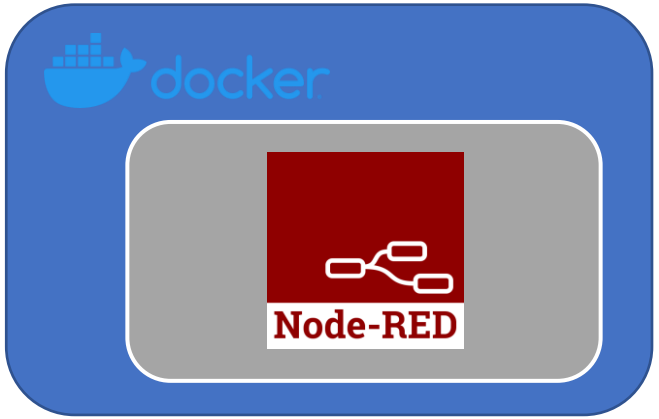
Scalability series part 1: Copying docker files via SD card

Mission:

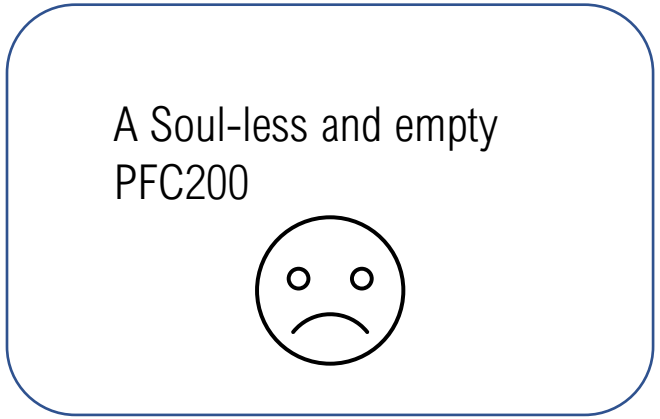
Create a way to quickly load an entire docker system from a running PFC (A) to a blank PFC (B) via SD card.

PFC A

PFC B



IP Address: 192.168.1.20



IP Address: 192.168.1.30

Scalability series part 1: Copying docker files via SD card

Hardware:

- PFC A
  - PFC200 Gen2 750-8212
  - 8DO module 750-530 (Any I/O mix is applicable)
  - End module 750-600
- PFC B
  - PFC200 Gen2 750-8212
  - 16DI module 750-1405 (Any I/O mix is applicable)
  - End module 750-600

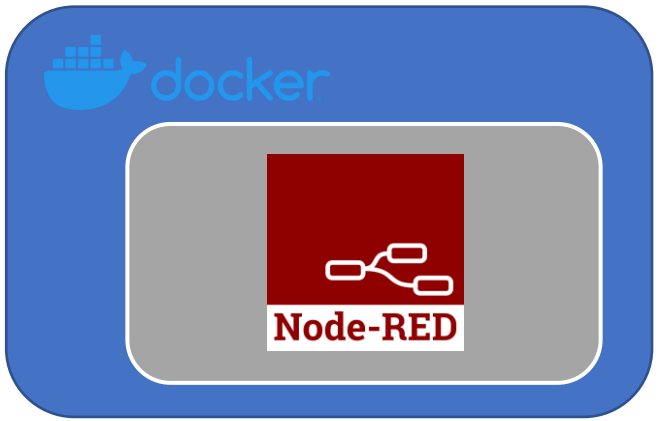
Firmware:

- PFC A
  - 3.09.05 (FW21)
  - Docker Version 20.10.5
- PFC B
  - 3.09.05 (FW21)

Software:

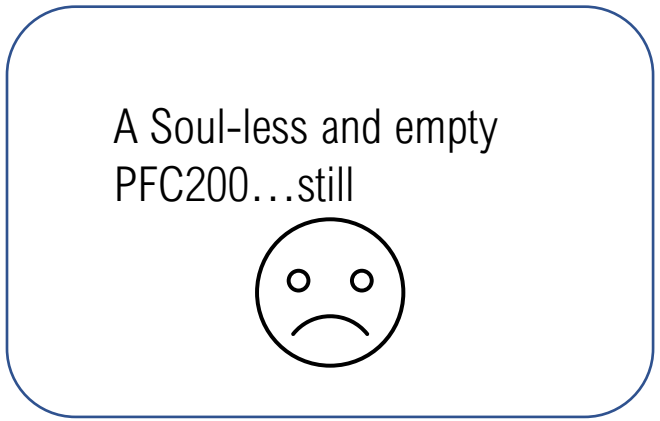
- PFC A
  - Node-RED (minimal) Version 3.0.2
- PFC B
  - None, nada, nothing

PFC A



IP Address: 192.168.1.20

PFC B



IP Address: 192.168.1.30

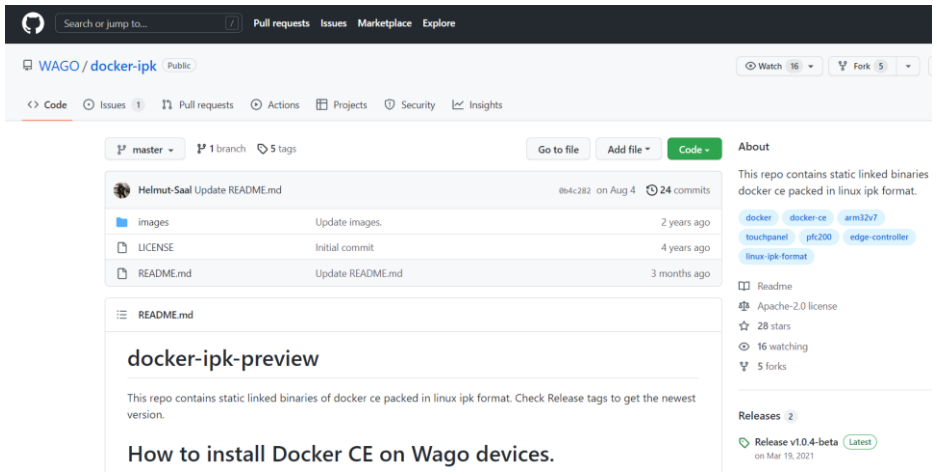
Scalability series part 1: Copying docker files via SD card

Step 1: Connect an empty SD card (8gb or 16gb) to your PC.

Step 2: Go to Wago's GitHub account and download the latest Docker IPK via https://github.com/WAGO/docker-ipk/releases

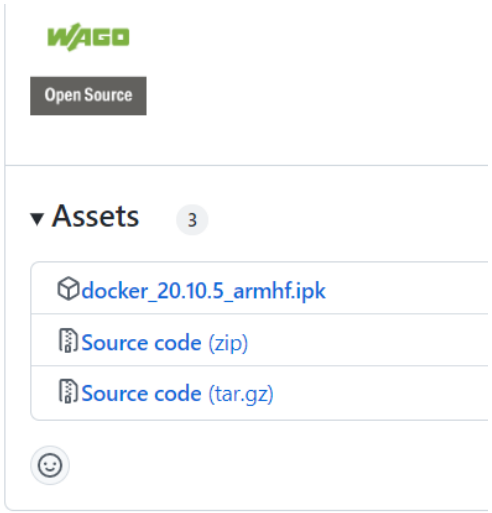
Step 3: Install the newly download IPK to the SD card

Step 4: Remove the SD card from your PC



GitHub

IP Address: 192.168.1.xx



Step 5: Insert your SD card with the docker ipk into your PFC and ssh into PFC A

Note: This SD card will need enough memory to store all docker images and the docker ipk (8gb to be safe).

Step 6: Once you've connected to PFC A, you can use the Docker save command to .tar the docker image (figure 2).

Docker save will save the image NOT the container, which houses all of the node-red flows. If you were to simply untar this new file, you will NOT retain your flow and palette data. We must extract the volume mounted to the container to do this.

Step 7: Find the volume associated with the docker container that you want to upload to the PFC B via docker inspect. This will give you access to the volume mount (figure 3).

Step 8: tar your docker volume

commands:
docker images
docker save <image name> > <new name>.tar
docker inspect <container ID>
tar cvf <new name>.tar <volume location>

Scalability series part 1: Copying docker files via SD card


```
root@PFC200V3-46216B:~ docker images
REPOSITORY          TAG                 IMAGE ID       CREATED         SIZE
nodered/node-red    latest-minimal      0b0f1c3f9ab8   2 months ago    223MB
root@PFC200V3-46216B:~ docker save nodered/node-red > nodered.tar
```

```
"Mounts": [
    {
        "Type": "volume",
        "Name": "node_red_user_data",
        "Source": "/home/docker/volumes/node_red_user_data/_data",
        "Destination": "/data",
        "Driver": "local",
        "Mode": "z",
        "RW": true,
        "Propagation": ""
    }
```

docker save nodered/node-red > nodered.tar

tar cvf node_red_volume.tar /home/docker/volumes/node_red_user_data

Step 9: Look into the root directory of PFC A and you should find several tar files. One containing your image backup and the other containing your volume backup.

Step 10: Use command df –h to look at the file structure in PFC A and find the SD card.

The SD card should be in the /media directory and the destination should look something like /media/<name_of_SD>

Step 11: Use the linux cp instruction to take all your tar files and bring them to the SD card.

Note: Make sure you're in the same directory as the tar files when using this command. This will ensure the correct source directory is used for the cp instruction

Step 12: Go into the SD card directory either through PFC A or computer and confirm the files are there (using cd instruction or PC)

Step 13: Once confirmed, move SD card to PFC B

commands:
df –h
cp <dockertar>.tar /media/<SD_card_name>
cd <directory_name>

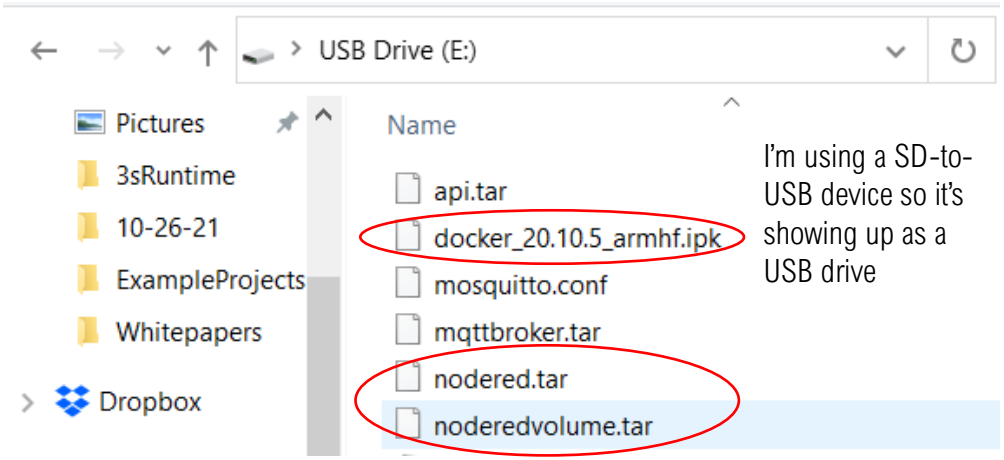
Scalability series part 1: Copying docker files via SD card

PFC A

PFC B

IP Address: 192.168.1.20

IP Address: 192.168.1.30



I'm using a SD-to-USB device so it's showing up as a USB drive

Step 14: When the SD card is inserted into PFC B, SSH into PFC B and use the terminal to go into the SD card directory.

Step 15: Use the docker load command to untar the docker image from the SD card.

Ex: docker load < nodered.tar

Step 16: Untar the nodered volume tar file. The volume will default its path to home/docker/volumes, so we will point the tar file to the root directory in order to follow that same path.

tar -xvf $(pwd)/noderedvolume.tar -C /

Step 17: Run a node-red container from the image and mount the newly imported volume to that image. You may need to change permissions to the volume by using chmod.

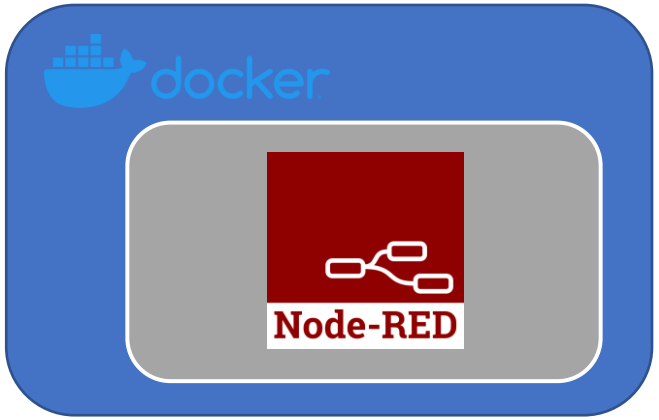
commands:
docker load < <name_of_docker_tar>
tar -xvf $(pwd)/<volume_name>.tar -C /

docker run –restart unless-stopped –d –p 1880:1880 --name <name_of_container> --network=host –security-opt seccomp:unconfined –v <volume_name> <image_name>

Scalability series part 1: Copying docker files via SD card

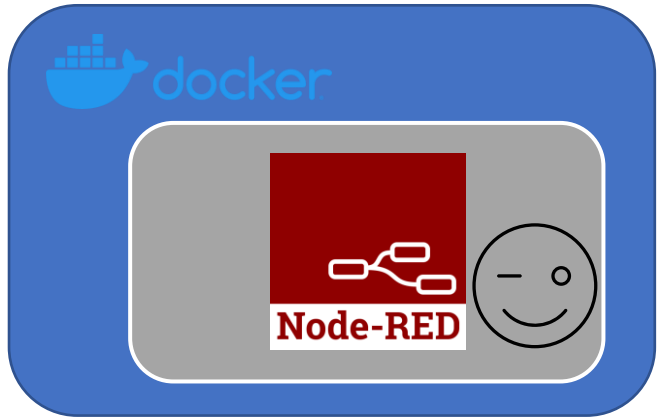
PFC A

IP Address: 192.168.1.20



PFC B

IP Address: 192.168.1.30



docker run --restart unless-stopped -d -p 1880:1880 --name node-red \--network=host --security-opt seccomp:unconfined \-v node_red_user_data:/data nodered/node-red:latest-minimal

Scalability series part 1: Copying docker files via SD card

What did we accomplish?

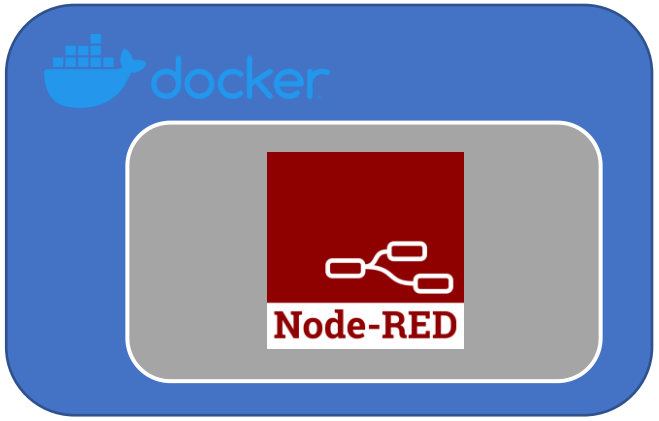
- ❑ Fit a new PFC with Docker as well as any subsequent Docker images/containers without an internet connection.
- ❑ Taking volumes from one container and mounting to another. This allows you to import any programs or settings from a container without needing to redo your work.
- ❑ Using a SD card as a transporting medium for easy access of your applications/programs.

What Next?

- ❑ Using a script or Dockerfile to automate the command line instructions.
- ❑ Create a magic SD card which will allow us to plug into the PFC and automatically execute these instructions.

PFC A

IP Address: 192.168.1.20



PFC B

IP Address: 192.168.1.30